Amendments to the Claims

Please amend the claims as follows:

1.   (Currently Amended) A method of converting data of a plurality of input data types to a plurality of output data types by an application program, said method comprising:

(a)   receiving a first attribute of a first input data type and a second attribute of a first output data type;

(b)   dynamically creating at runtime a first optimized conversion routine based on said first attribute and said second attribute, the conversion routine including one or more computer instructions to be executed during conversion; ~~and~~

(c)   validating specific field conversion options of the conversion routine; and

(d) ~~(c)~~ executing said first optimized conversion routine from said application program to convert data of said first input data type to said first output data type~~, the first output data type not being dependent on the first input data type~~.

2.   (Currently Amended) The method of claim 1, wherein step (d) ~~(c)~~ comprises calling said first optimized conversion routine from said application.

3.   (Currently Amended) The method of claim 1, wherein step (d) ~~(c)~~ comprises storing said first optimized conversion routine inline with said application.

4.   (Original) The method of claim 1, wherein step (b) is performed dynamically while said application program is executing.

5.   (Currently Amended) The method of claim 1, further comprising:

(e) ~~(d)~~ receiving a third attribute of a second input data type and a fourth attribute of a second output data type;

(f) ~~(e)~~ generating a second optimized conversion routine based on said third attribute and said fourth attribute; and

(g) ~~(f)~~ executing said second optimized conversion routine from said application program to convert input data of said second input data type to said second output data type.

6.   (Original) The method of claim 1, wherein said first and second attribute is character type.

7.   (Original) The method of claim 1, further comprising generating program debugging instrumentation for said first optimized conversion routine.

8.   (Currently Amended) A method of converting data from input data types to output data types, said method comprising:

(a)   receiving a plurality of input attributes and output attributes from an application program;

(b)   dynamically creating at runtime a plurality of data conversion routines for each set of input attributes and output attributes, the data conversion routines for converting data from one data type having an input attribute to another data type having an output attribute, the one data type having an input attribute not being dependent on the other data types having an output attribute, the conversion routines including one or more computer instructions to be executed during conversion; and

(c)   determining the size of the one or more computer instructions;

(d)   determining whether the one or more data conversion routines should be generated as stand-alone routines or code chunks based on the size determination; and

(e) (c) storing said plurality of data conversion routines in memory accessible to said application program.

9.   (Previously Presented) The method of claim 8, wherein said data conversion routines are callable by said application program.

10.   (Previously Presented) The method of claim 8, wherein said data conversion routines are stored inline in said application program.

11.   (Previously Presented) The method of claim 8, wherein step (b) is performed dynamically while said application program is executing.

12. (Original) The method of claim 8, wherein said input and output attributes are character type.

13. (Original) The method of claim 8, wherein said input and output attributes are date type.

14. (Previously Presented) The method of claim 8, further comprising generating program debugging instrumentation for said plurality of data conversion routines.

15. (Currently Amended) A system for dynamically generating computer data conversion routines, said system comprising:

a processor; and

a memory device coupled to said processor;

wherein said system is adapted to receive a plurality of input attributes and output attributes from an application program; and

wherein said memory device stores instructions that, when executed by said processor, cause said processor to:

dynamically create at runtime a plurality of data type conversion routines for each set of input attributes and output attributes, the conversion routines including one or more computer instructions to be executed during conversion; and

validate specific field conversion options of the conversion routines;

determine the size of the one or more computer instructions;

determine whether the conversion routines should be generated as stand-alone routines or code chunks based on the size determination; and

store said plurality of data conversion routines in a second memory device accessible to said application program, the data conversion routines for converting data from one data type having an input attribute to another data type having an output attribute;

wherein the another data type having an output attribute is not dependent on the one data type having an input attribute.

16.   (Canceled)

17.   (Canceled)

18.   (Previously Presented) The system of claim 15, wherein said plurality of data conversion routines are generated while said application program is executing.

19.   (Original) The system of claim 15, wherein said input attributes are character type and said output attributes are date type.

20.   (Previously Presented) The system of claim 15, wherein said processor further generates program debugging instrumentation for said plurality of data conversion routines.